UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
08/831,845	04/01/1997	BARTLEY H. CALDER	P2167/SUN1P1	9132

22434	7590	11/20/2003

BEYER WEAVER & THOMAS LLP
P.O. BOX 778
BERKELEY, CA 94704-0778

EXAMINER
BULLOCK JR, LEWIS ALEXANDER

ART UNIT	PAPER NUMBER
2126	34

DATE MAILED: 11/20/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 08/831,845	**Applicant(s)** CALDER ET AL.
	Examiner Lewis A. Bullock, Jr.	**Art Unit** 2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE <u>3</u> MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on <u>*8/11/03*</u>.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

   4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

   Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

   Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   a)☐ All b)☐ Some * c)☐ None of:

   1.☐ Certified copies of the priority documents have been received.

   2.☐ Certified copies of the priority documents have been received in Application No. _____.

   3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

   * See the attached detailed Office action for a list of the certified copies not received.

13)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

   a) ☐ The translation of the foreign language provisional application has been received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____.

4) ☐ Interview Summary (PTO-413) Paper No(s). _____.

5) ☐ Notice of Informal Patent Application (PTO-152)

6) ☐ Other: .

DETAILED ACTION

1.   The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all

obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set
> forth in section 102 of this title, if the differences between the subject matter sought to be patented and
> the prior art are such that the subject matter as a whole would have been obvious at the time the
> invention was made to a person having ordinary skill in the art to which said subject matter pertains.
> Patentability shall not be negatived by the manner in which the invention was made.

2.   Claims 1-27 are rejected under 35 U.S.C. 103(a) as being unpatentable over

GOSLING (EP 718761 A1) in view of PASHUPATHY (US 6,078,951).

As to claim 1, GOSLING teaches a computer-implemented framework for

associating data (object) with a command object (viewer) wherein the data is associated

with an application (user interface control program / user's web access program), the

computer-implemented framework comprising: a data handler mechanism (class loader

/ inter-computer link control program); a data retriever mechanism (objects directory /

disc directory or catalog / objects); and a mapping mechanism (viewer library) (pg. 2,

lines 37-44; pg. 4, lines 11-22, pg. 5, lines 24-35; pg. 5, lines 49-58; pg. 6, lines 15-28;

pg. 7, lines 8-9; pg. 24; lines 6-24). However, GOSLING does not explicitly mention that

the data handler mechanism allows use of new command objects without modifying the

application. GOSLING does teach that invention allows object viewers unknown, herein

new to the user's web access program to be dynamically used and added to the system

in a viewer library for the user (pg. 2, line 37 – pg. 3, line 11). It would be inherent that

this dynamic loading does not modifying the users web access program since the

viewer library is changed by the addition of the new viewer and not the web access

program. However, GOSLING does not teach returning the command list for display.

PASHUPATHY teaches a mapping mechanism (server container) being

arranged to obtain a command list (list of viewers for the file type) for the application to

display (user has to select appropriate viewer if plural viewers are capable of accessing

the file), the command list (list of viewers for the file type) including the command object

(user-selected viewer) obtained, wherein the command object is executed by the

application upon selection of the command object from the command list (col. 6, lines

62 – col. 7, line 4). It is inherent that the viewers returned are displayed in a list since

the user must select the appropriate viewer to use to view the data. It is also obvious

based on the combination of GOSLING in view of PASHUPATHY that the command list

is returned to the data handler mechanism (class loader) then the application (user's

web access program) for displaying the command list in order to receive the subsequent

selection by the user. Therefore, it would be obvious to combine the teachings of

GOSLING with the teachings of PASHUPATHY in order to automate the searching,

installation, configuration, and updating of software for a computer system (col. 1, line

58-60).

As to claim 2, GOSLING teaches the downloading of data and binding such data

to a command object (pg. 6, lines 15-28). It would be obvious that since the handle is

initially received prior to the body that the data is a stream of bytes over the network.

As to claim 3, GOSLING teaches that the two computer systems have different

computer platforms and a variety of operating systems (pg. 4, lines 23-28). It would be

obvious that that there would have to exist a mechanism for converting the data from one format understandable by one platform to another in order for data to be processed and interpreted for viewing.

As to claim 4, GOSLING teaches the server computer system is a Sun Microsystems computer (pg. 4, line 25). It would be obvious that since the data object and the command object are retrieved from the server computer system and since it is well known in the art that a Sun system is formulated in Java that the data object and command object are created in the Java programming language.

As to claim 5, GOSLING teaches the data is text data (pg. 5, lines 28-35).

As to claim 6, GOSLING teaches the data handler is arranged to receive a request from the application, to bind the data to the command object, and to return the command object to the application (pg. 6, lines 15-28; pg. 7, lines 8-9).

As to claim 7, GOSLING teaches that the two computer systems have different computer platforms and a variety of operating systems (pg. 4, lines 23-28). It would be obvious that that there would have to exist a mechanism for converting the data from one format understandable by one platform to another in order for data to be processed and interpreted for viewing.

As to claim 8, GOSLING teaches the mapping mechanism includes a look-up table (listing) arranged to associate the command object with the data (pg. 6, lines 22-28).

As to claim 9, GOSLING teaches a computer implemented method for associating data (object) with a command object (viewer) in response to a request from an application (user interface control program / user's web access program), the method comprising: accessing the data through an interface (class loader / inter-computer link control program) in response to the request from the application; accessing a mapping mechanism (viewer library) which is independent of the interface but in communication with the interface to locate an installed command object (viewer known to user) that is appropriate for the data; obtaining the command object; binding the command object to the data; and returning the command object to the application to allow execution of the command object (viewer allows object to be accessible to user) (pg. 2, lines 37-44; pg. 4, lines 11-22, pg. 5, lines 24-35; pg. 5, lines 49-58; pg. 6, lines 15-28; pg. 7, lines 8-9; pg. 24; lines 6-24). However, GOSLING does not explicitly mention that the interface allows use of new command objects without modifying the application. GOSLING does teach that invention allows object viewers unknown, herein new to the user's web access program to be dynamically used and added to the system in a viewer library for the user (pg. 2, line 37 – pg. 3, line 11). It would be inherent that this dynamic loading does not modifying the users web access program since the

viewer library is changed by the addition of the new viewer and not the web access program. However, GOSLING does not teach returning the command list for display.

PASHUPATHY teaches obtaining a command list identifying commands (names of viewers) associated with the data (file type found) and wherein the command list (names of viewers) is returned for display (col. 6, lines 62 – col. 7, line 4).). It is inherent that the viewers returned are displayed in a list since the user must select the appropriate viewer to use to view the data. It is also obvious based on the combination of GOSLING in view of PASHUPATHY that the command list is returned to the data handler mechanism (class loader) then the application (user's web access program) for displaying the command list in order to receive the subsequent selection by the user. Therefore, it would be obvious to combine the teachings of GOSLING with the teachings of PASHUPATHY in order to automate the searching, installation, configuration, and updating of software for a computer system (col. 1, line 58-60).

As to claim 10, GOSLING teaches that the two computer systems have different computer platforms and a variety of operating systems (pg. 4, lines 23-28). It would be obvious that that there would have to exist a mechanism for converting the data from one format understandable by one platform to another in order for data to be processed and interpreted for viewing.

As to claim 11, GOSLING teaches the server computer system is a Sun Microsystems computer (pg. 4, line 25). It would be obvious that since the data object

and the command object are retrieved from the server computer system and since it is

well known in the art that a Sun system is formulated in Java that the data object and

command object are created in the Java programming language.

As to claim 12, GOSLING teaches the downloading of data and binding such

data to a command object (pg. 6, lines 15-28). It would be obvious that since the

handle is initially received prior to the body that the data is a stream of bytes over the

network.

As to claim 13, GOSLING teaches operating on the data using the command

object (viewing the object) (pg. 7, lines 8-9).

As to claim 14, GOSLING teaches the command object is selected from a set of

command objects associated with a command list (listing within the viewer library) and

accessing the command list through the interface (pg. 6, lines 22-28).

As to claim 15, PASHUPATHY receiving a request for a command list from the

application (check to see if viewer is most current version) obtaining the command list

(names of viewers) associated with a type of data (file type found) and wherein the

command list (names of viewers) is returned for display (col. 6, lines 62 – col. 7, line 4).

It is inherent that the viewers returned are displayed in a list. It is also obvious based on

the combination of GOSLING in view of PASHUPATHY that the command list is

returned to the interface (class loader / inter-computer link control program) that performs the cited steps and then the application (user's web access program) for display and subsequent selection by the user. Therefore, it would be obvious to combine the teachings of GOSLING with the teachings of PASHUPATHY in order to automate the searching, installation, configuration, and updating of software for a computer system (col. 1, line 58-60).

As to claims 16-20, reference is made to a computer program product which corresponds to the method of claims 9-11, 13, and 14 and is therefore met by the rejection of claims 9-11, 13, and 14 above. Claim 16 also details the mapping mechanism is not a part of the application. GOSLING teaches the mapping mechanism is not part of the application (figures 2 & 3, viewer library is on server also).

As to claim 21, GOSLING teaches the command object (viewer) is obtained by the mapping mechanism (viewer library) based substantially on the data (handle) without an external input from a user of the application (pg. 6, lines 15-28; pg. 3, lines 1-9).

As to claim 22, GOSLING teaches the command object (viewer) is obtained by the mapping mechanism (viewer library) based substantially on the data (handle) without directly involving the application (pg. 6, lines 15-28; pg. 3, lines 1-9).

As to claim 23, refer to claim 1 for rejection. However, claim 23 further details the data handler mechanism is independent and interfacing with a plurality of applications. GOSLING teaches the client system may be a variety of different computer platforms and a variety of operating systems (pg. 4, lines 27-28). Multitasking operating systems having a executing a plurality of applications are well known in the art and therefore would be obvious in view of the teachings of GOSLING.

As to claim 25, GOSLING teaches the mapping mechanism (viewer library) is not a component of the data handler mechanism (class loader / inter-computer link control program) (fig 3).

As to claims 24 and 26, GOSLING teaches the mapping mechanism (viewer library) and the data handler mechanism (class loader / inter-computer link control program) are separately maintained (fig 3).

As to claim 27, GOSLING teaches that the client system and the server system are two different computer platforms (pg. 4, lines 23-28). It would be obvious then that the application and class loader are the specific to one another since they are on the same system while the viewer library is not specific to the application since it is on a different system.

Response to Arguments

3.   Applicant's arguments filed 8/11/03 have been fully considered but they are not

persuasive. Applicant argues that none of the references teach the data handler

mechanism allowing for the use of new command objects without modifying the

application. The examiner disagrees. Although, neither of the combination of Gosling

and Pashupathy used explicitly language that new command objects are used without

modifying the application, the teachings of Gosling and Pasupathy inherently teach this

limitation. Gosling and Pasupathy each teach that viewers (command objects) unknown

to a client system are downloaded and accessible to an application in order to access

data. Gosling teaches this at page 6, lines 29-57, wherein if the appropriate viewer

(command object) is not located within the viewer library, indicating that the selected

object is of a data type which is unfamiliar to the user workstation such that no viewer is

stored to access it, the class loader executes a search for an appropriate viewer and

downloads the viewer if it is found. In addition, since the user is accessing the system

through a web access program (pg. 2, line 36-44) and the viewer is found and

downloaded to enable access to the data, the downloaded command object is otherwise

new to the web access program. Pasupathy also teaches that if a viewer that supports

the file type is not found, the client computer system interfaces with a server computer

system, the names of viewers that support the file type is provided to the client

computer system, the user of the client computer system is prompted to select one of

the viewers provided, and the client computer system downloads the selected viewer to

enable access to a selected file (col. 6, line 49 – col. 7, line 25). Therefore, the viewer

downloaded is initially unknown and subsequently new to the client system when it is found and loaded for the user to access the file. In addition, since browser applications allow users to interface with the Internet in order to display files that are in many different formats by using viewers (col. 1, line 16-57) and the client computer system downloads the appropriate viewer when it is unknown making the viewer subsequently new to the client, that the viewer is new to the browser which allows the user to access the remote data. Therefore, as indicated in the rejection the combination teach a generic interface (class loader) which allows for the use of new command objects (viewers unknown to the browser but subsequently founded and downloaded). The claim then states that this use is performed without modifying the application. In the recitations listed above for both reference, modification (addition) for a new command object, i.e. viewer, is only done to the client's list such that new viewer is added to a list that the application, i.e. browser, has access to and/or the operating system. There is no language in either prior art of record wherein the modification is made to the application. Therefore, the prior art of record allows for the use of new command objects without modifying the application. It is noted in Applicant's specification wherein the invention enables an application to be "blind" to the process of binding data with an appropriate command object and thereby allows new types of data and therefore, new command objects, to be created for use with the application without requiring that the application be **modified to recognize** the new command objects (col. 7, lines 16-28). Applicant's specification states prior techniques that an application recognizes a command object was by a look-up table used by the application to associate data with

command objects wherein new command objects modify the list that the application use

(col. 3, lines 5-10). The prior art techniques of Gosling and Pasupathy allow for the list

to be modified when a new viewer is discovered, i.e. found, but the application is never

modified. Therefore, the prior art techniques meet the claimed limitations as disclosed

wherein the application itself is not directly modified, but does not meet Applicant's

concept of **allowing the use of new command objects without modifying the**

application to recognize the new command objects wherein a list used by the

application is not modified as defined in the specification.

Applicant then argues that the combination does not teach or suggest a

command map obtaining a command list, providing the command list to the application

by using a data handler, and displaying the command list. The examiner disagrees.

Gosling teaches a data handler (class loader) that receives commands (via hyperlinks)

from an application (browser / users web access program) and downloads data (object)

as well as an appropriate command object (viewer) to enable the application to access

the command object and data (pg. 2, line 33-53; pg. 5, line 49 – pg. 6, line 9). Gosling

also teaches that before viewers are downloaded, a determination is made whether or

not to accept the object viewer. In order to perform the determination, the user is

asked whether or not to accept the object viewer or a default decision to accept or not

accept is initiated (pg. 6, lines 43-48). Pashupathy teaches a client computer system

determines whether a command object (viewer) that supports a format or type of data

(file) is locally known and if not retrieves the names of viewers that support the file type

from a server computer system and provides those names to the user on the client

computer system for selection of one of the viewers wherein upon such selection the

viewer is downloaded and installed to display the data (col. 6, line 35 – col. 7, line 25).

Pashupathy makes no mention of what entity of the client computer system is

performing the cited determination and obtaining. Therefore, based on the combination

of Gosling with Pashupathy the class loader of the client computer system retrieves a

list names of viewers that support the file type from a server computer system, i.e.

command map, and provides those names to the user, via the browser (application)

such that upon selection of a viewer the viewer is downloaded and installed to display

the data. Hence, the combination of Gosling with Pashupathy teaches or suggest a

command map (server computer system) obtaining a command list (list names of

viewers), providing the command list (list names of viewers) to the application (browser /

user web access program) by using a data handler (class loader), and displaying the

command list (for selection of an appropriate viewer by the user). Therefore, the

combination is adequately met by the claims as disclosed above.

Applicant then argues that the independent claims variably recite a command list

including a command object for execution by an application. In responding the

examiner would like to point out that Applicant's claimed execution of the command

object is similar to Gosling's selection of a viewer and the execution of the selected

viewer that enables the object or data to be accessible or displayed. Applicant then

states that the techniques of the present invention recognize that many command

objects should all be available to many different applications wherein in one example,

"an application is arranged to utilize a set of command objects that are used by

numerous applications." In responding, the examiner would like to point out that none of the claims disclose the command objects are usable by numerous applications. The closest claim to what Applicant is stating is claim 23, however, claim 23 as disclosed does not allude to this statement. Claim 23, details that the data handler mechanism is interfacing with a plurality of applications wherein a selected application has access to new command objects and that the mapping mechanism is associated with the plurality of applications. Hence, claim 23 broadly interpreted allows for one of a plurality of applications to access new command objects via a data handler mechanism and a mapping mechanism, wherein the other applications would never use command objects. Therefore, none of the claims specifically detail the command objects are usable by numerous applications. In addition, Gosling teaches that user node or client node may utilize different computer platforms and operating systems (col. 4, line 36-39; col. 4, lines 27-29). The examiner has stated that multitasking operating systems wherein a plurality of applications execute on an computer system is well known in the art and therefore would be obvious to one skilled in the art to have a multitasking operating system. The examiner has cited other prior art of record to show that a multitasking operating system that executes a plurality of programs is well known in the art. Gosling teaches a user access program. Therefore, it would be obvious that one of the programs is a user access program. It would also be obvious to one skilled in the art that the other applications are various other applications. Therefore, the examiner believes the limitation is broadly met and therefore, the claims are rejected as disclosed above.

Conclusion

4.   **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time

policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE

MONTHS from the mailing date of this action. In the event a first reply is filed within

TWO MONTHS of the mailing date of this final action and the advisory action is not

mailed until after the end of the THREE-MONTH shortened statutory period, then the

shortened statutory period will expire on the date the advisory action is mailed, and any

extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of

the advisory action. In no event, however, will the statutory period for reply expire later

than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the

examiner should be directed to Lewis A. Bullock, Jr. whose telephone number is (703)

305-0439. The examiner can normally be reached on Monday-Friday, 8:30 am - 5:00

pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's

supervisor, John A Follansbee can be reached on (703) 305-8498. The fax phone

number for the organization where this application or proceeding is assigned is (703)

746-7239.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-0286.

lab

JOHN FOLLANSBEE
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100